Discrete Mathematics 119 (1993) 13–20
North-Holland

An algorithm for optimum common root functions of two digraphs*

Mao-cheng Cai
Institute of Systems Science, Academia Sinica, Beijing 100080, China

Received 25 June 1990
Revised 22 October 1991

Abstract

Cai, M., An algorithm for optimum common root functions of two digraphs, Discrete Mathematics 119 (1993) 13–20.

Let G_1 and G_2 be finite digraphs, both with vertex set V. Suppose that each v of V has nonnegative integers $f(v)$ and $g(v)$ with $f(v) \leqslant g(v)$, and each arc e of G_i has nonnegative integers $a_i(e)$ and $b_i(e)$ with $a_i(e) \leqslant b_i(e)$, $i=1,2$. In Cai (1990) a necessary and sufficient condition was given for the existence of k arborescences in G_i covering each arc e of G_i at least $a_i(e)$ and at most $b_i(e)$ times, $i=1,2$, and satisfying the condition that for each v in V

$$f(v) \leqslant r_1(v) = r_2(v) \leqslant g(v),$$

where $r_i(v)$ denotes the number of the arborescences in G_i rooted at v. Such an r_i is called a common root function and denoted by r.

In this paper, we present a polynomial algorithm for finding an optimum common root function r for a given weight function defined on V.

Let $G=(V, A)$ be a finite digraph with vertex set V and arc set A and without loops. We write $\bar{S}=V \backslash S$ for $S \subseteq V$ and $V \backslash v = V \backslash \{v\}$ for $v \in V$. An arc from vertex u to v is denoted by uv. For disjoint $S, T \subseteq V$, let $A(S, T)$ denote the set of arcs of G from S to T, $D_G^-(S)=A(\bar{S}, S)$, $d_G^-(S)=|D_G^-(S)|$, $D_G^+(S)=D_G^-(\bar{S})$ and $d_G^+(S)=|D_G^+(S)|$.

An *arborescence* of G is defined as a spanning tree directed in such a way that each vertex of G, except one, called the root of the arborescence, has one arc entering it. We say that an arc subset B of $G=(V, A)$ is covered by arborescences $T_1, \ldots, T_k$ if every arc of B is in at least one T_i.

Correspondence to: Mao-cheng Cai, Institute of Systems Science, Academia Sinica, Beijing 100080, People's Republic of China.

* Research partially supported by Rutgers University's Center for Operations Research, by the Air Force Office for Scientific Research (grant #AFOSR-89-0512 and #AFOSR-90-0008), and by the National Science Foundation (grant #NSF-DMS-8906870). Work was done while the author was visiting Rutgers University's Center for Operations Research, March–June, 1990.

For any function f defined on S and $T \subseteq S$, where S may denote either a subset of vertices or a subset of arcs, we write $f(T) = \sum_{t \in T} f(t)$, and set $f(\emptyset) = 0$. Let $\mathbf{Z}_+$ denote the set of nonnegative integers.

Definition 1.1. Let $G_1 = (V, A_1)$ and $G_2 = (V, A_2)$ be two finite digraphs, both with vertex set V, let f and g be given functions: $V \to \mathbf{Z}_+$ such that $f \leqslant g$, and let a_i and b_i be functions: $A_i \to \mathbf{Z}_+$ with $a_i \leqslant b_i$, $i = 1, 2$. A function $r: V \to \mathbf{Z}_+$ is called a common root function of G_1 and G_2 if there exist $r(V)$ (not necessarily distinct) arborescences in G_i covering each arc e of G_i at least $a_i(e)$ and at most $b_i(e)$ times, $i = 1, 2$, and satisfying the condition that for each $v \in V$

$$f(v) \leqslant r_i(v) = r(v) \leqslant g(v),$$

where r_i denotes the number of the arborescences which are rooted at v, $i = 1, 2$.

In [3] we proved the following theorem, which is a generalization and unification of most packing and covering theorems concerning arborescences.

Theorem 1.2. *Let G_1, G_2, f, g, a_i and b_i, $i = 1, 2$, be given as in Definition 1.1, and k be a positive integer. Then G_1 and G_2 have a common root function r, with $r(V) = k$ if and only if for any two families $\mathscr{F}_1$ and $\mathscr{F}_2$ (possibly empty) of disjoint (non-empty) subsets of V, and any $I_i \subseteq \bigcup_{S \in \mathscr{F}_i} S$, $i = 1, 2$,*

$$\sum_{i=1}^{2} \sum_{S \in \mathscr{F}_i} \left\{ k - \sum_{v \in S \cap I_i} b_i(D^-_{G_i}(v) \cap D^-_{G_i}(S)) - \sum_{v \in S \setminus I_i} [k - a_i(D^-_{G_i}(v) \setminus D^-_{G_i}(S))] \right\}$$
$$- g(I_1 \cap I_2) + f(\overline{I_1 \cup I_2}) \leqslant k. \tag{1}$$

Clearly, by choosing suitable parameters, most of packing and covering theorems concerning arborescences can easily be obtained from the theorem as special cases.

The present paper is a natural continuation of [3]. Let $G_1, G_2, f, g, a_1, b_1, a_2, b_2$ and k be given as in Theorem 1.2, and w be a real weight function defined on V. The purpose of this paper is to present a polynomial algorithm for finding a minimum weighted common root function r of G_1 and G_2 with $r(V) = k$.

We may assume, without loss of generality, that, for every $v \in V$, $i = 1, 2$,

$$a_i(D^-_{G_i}(v)) + f(v) \leqslant k. \tag{2}$$

Indeed, if G_i has k required arborescences, obviously (2) holds. On the other hand, by taking $\mathscr{F}_i = \{V\}$, $\mathscr{F}_{3-i} = \emptyset$ and $I_i = V \setminus v$, then (2) yields (1).

For $i = 1, 2$, we construct an auxiliary digraph $G_i^* = (V^*, A_i^*)$ from G_i as follows. Let $V' = \{v' \colon v \in V\}$ and $V^* = V \cup V' \cup \{s\}$ (where the v' are new vertices and s is a new specified vertex, called a source), and let A_i^* consist of the following type of arcs:

$$uv,\ uv' \in A_i^* \quad \text{if } uv \in A_i,$$

$$sv,\ sv',\ vv',\ v'v \in A_i^* \quad \text{for every } v \in V.$$

For each $v' \in V'$, let

$$b(v') = g(v) - f(v)$$

and for each arc $e \in A_i^*$, $i = 1, 2$, let

$$b_i^*(e) = \begin{cases} a_i(e) & \text{if } e \in A_i, \\ b_i(e) - a_i(e) & \text{if } e = uv' \text{ and } uv \in A_i, \\ k & \text{if } e = vv', \\ k - f(v) - \sum_{e \in D_{G_i}^-(v)} a_i(e) & \text{if } e = v'v, \\ f(v) & \text{if } e = sv, \\ b(v') & \text{if } e = sv'. \end{cases}$$

Then it follows from the construction of G_i^* and the definition of $b_i^*(e)$ that G_i has k (not necessarily distinct) arborescences covering each arc $e \in A_i$ at least $a_i(e)$ and at most $b_i(e)$ times and satisfying $f(v) \leqslant r_i(v) \leqslant g(v)$ for each $v \in V$, where $r_i(v)$ denotes the number of the arborescences rooted at v, if and only if G_i^* has k (not necessarily distinct) arborescences rooted at s covering each $e \in A_i^*$ at most b_i^* times, each of them containing exactly one arc leaving s.

Note that, for each $v \in V$, $b_i^*(D_{G_i^*}^-(v)) = k$, then each $e \in D_{G_i^*}^-(v)$, $v \in V$, is covered exactly b_i^* times. For each $v' \in V'$, set weight $w^*(v') = w(v)$, and for each $e \in A_i^* \setminus A_i^*(s, V')$, set its capacity

$$c_i(e) = b_i^*(e).$$

Then it is easily seen that to find the required common root function r of G_1 and G_2 is equivalent to find an integer optimum solution to the following linear programming L:

$$\min \sum_{v' \in V'} w^*(v') h(v') \tag{3}$$

$$L: \begin{array}{l} 0 \leqslant h(v') \leqslant b(v') \text{ for all } v' \in V', \\ h(X \cap V') \geqslant k - c_i(D_{G_i^*}^-(X) \setminus A_i^*(s, V')) \\ \qquad \text{for all } \emptyset \neq X \subseteq V \cup V', \quad i = 1, 2, \qquad (4) \\ h(V') = k - f(V) \qquad (5) \end{array}$$

(see [1] for more details).

If we think of $h(v')$ as the capacity $c_i(sv')$ of arc $sv' \in A_i^*(s, V')$, then (5) is equivalent to

$$c_i(D_{G_i^*}^+(s)) = k, \tag{6}$$

and (4) to

$$c_i(D_{G_i^*}^-(X)) \geqslant k \quad \text{for all } \emptyset \neq X \subseteq V \cup V', \ i = 1, 2. \tag{7}$$

Now we present the following algorithm, which is an adaptation of the algorithm given in [5].

Algorithm

Step 1

1.0. For $i=1,2$, construct the auxiliary digraph $G_i^*=(V^*, A_i^*)$ from G_i.

1.1. For each $v'\in V'$, let $b(v')=g(v)-f(v)$, weight $w^*(v')=w(v)$.

1.2. For each $e\in A_i^*$, $i=1,2$, define $b_i^*(e)$.

Step 2

2.0. Let $w_1=0$ and $w_2=w^*$.

2.1. For each $v'\in V'$, set $h(v')=b(v')$.

2.2. For each $e\in A_i^*\setminus A_i^*(s, V')$, $i=1,2$, set capacity $c_i(e)=b_i^*(e)$.

Step 3

3.0. For each $v'\in V'$ and $i=1,2$, set capacity $c_i(sv')=h(v')$.

3.1. Applying a max-flow min-cut algorithm, for each $v'\in V'$ and $i=1,2$, determine the minimum subset $S_i(v')$ such that $v'\in S_i(v')\subseteq V\cup V'$ and $c_i(D_{G_i^*}^-(S_i(v')))$ is minimum. If, for some v', $c_i(D_{G_i^*}^-(S_i(v')))<k$, then the required common root function does not exist. HALT.

Step 4

4.0. Construct the auxiliary digraph $H=(V', A)$.

4.1. For $i=1,2$, determine m_i and T_i.

4.2. Find the shortest path P from T_2 to T_1 in H by the labelling technique, using the labels having defined but not deleted previously. If P exists, go to Step 6.

Step 5

5.0. Let B denote the set of vertices in H having labels.

5.1. Count $\delta_1, \delta_2, \delta_3, \delta_4$ and δ.

5.2. If δ is finite, then modify $w_1(x)$ and $w_2(x)$, for all $x\in B$, and go to Step 4. Otherwise, if $h(V')=k-f(V)$, then r defined by $r(v)=h(v')+f(v)$ for all $v\in V$ is a minimum weighted common root function, HALT; otherwise, construct the two families $\mathscr{F}_1$ and $\mathscr{F}_2$ of disjoint (nonempty) subsets of V and $I_i\subseteq\cup_{S\in\mathscr{F}_i}S$, $i=1,2$, for which (1) does not hold, G_1 and G_2 have no common root function r with $r(V)=k$, HALT.

Step 6

6.0. Count $\varepsilon_1, \varepsilon_2$ and ε.

6.1. Modify $h(x)$ for each x on P.

6.2. Delete all the labels.

6.3. Go to Step 3.

Now we describe the algorithm in some details and verify its correctness.

Let $L(\lambda)$ denote the linear programming obtained from L by replacing (5) by

$$h(V')=\lambda$$

for some integer λ such that $k-f(V)\leqslant\lambda\leqslant b(V')$. Then $L=L(k-f(V))$. At the beginning, for each $v'\in V'$, set

$$h(v')=b(v').$$

Clearly, (3) holds. We may assume (4) also holds, for otherwise G_1 and G_2 have no common root function r with $r(V)=k$. Therefore $h(V')\geqslant k-f(V)$ and h is an integer optimum solution to $L(h(V'))$. If $h(V')=k-f(V)$, then $h=b=g-f$ is a required solution. So we assume $h(V')>k-f(V)$.

Split weight w^* into $w_1=0$ and $w_2=w^*-w_1$. Then h is an integer optimum solution to linear programming $L(\lambda, w_i)$ for $\lambda=h(V')$:

$$\min \sum_{v'\in V'} w_i(v')h(v')$$

$$h(v')\leqslant b(v') \text{ for all } v'\in V',$$

$$L(\lambda, w_i)\text{: } h(X\cap V')\geqslant k-c_i(D^-_{G_i^*}(X)\setminus A_i^*(s, V')) \text{ for all } \emptyset\neq X\subseteq V\cup V',$$

$$h(V')=\lambda.$$

From h, w_1 and w_2, we construct new h', w'_1 and w'_2 such that $w'_1+w'_2=w^*$ and h' is an integer optimum solution to $L_i(\lambda', w'_i)$, $i=1,2$, for some integer λ' such that $k-f(V)\leqslant\lambda'\leqslant h(V')$.

The procedure proceeds as follows.

For each $v'\in V'$ and $i=1,2$, set the capacity

$$c_i(sv')=h(v').$$

Then (7) holds. Applying a max-flow min-cut algorithm, determine the minimum subset $S_i(v')$ such that $v'\in S_i(v')\subseteq V\cup V'$ and $c_i(D^-_{G_i^*}\, S_i(v')))$ is minimum.

For $i=1,2$, set

$$m_i=\max\{w_i(x)\text{: } x\in V',\ h(x)>0,\ c_iD^-_{G_i^*}(S_i(x)))>k\},$$

$$T_i=\{x\in V'\text{: } h(x)>0,\ c_i(D^-_{G_i^*}(S_i(x)))>k,\ w_i(x)=m_i\}.$$

Make an auxiliary digraph $H=(V', A)$ as follows.

- If $h(x)>0$, $c_1(D^-_{G_1^*}(S_1(x)))=k$, $y\in S_1(x)$, $h(y)<b(y)$ and $w_1(y)=w_1(x)$, then let $xy\in A$.
- If $h(x)>0$, $c_2(D^-_{G_2^*}(S_2(x)))=k$, $y\in S_2(x)$, $h(y)<b(y)$ and $w_2(y)=w_2(x)$, then let $yx\in A$.

By the labelling technique, decide whether there exists a path from the subset T_2 to T_1 in H.

Case 1: There is a path from T_2 to T_1.

Let $P=x_1x_2\cdots x_{2l+1}$ be the shortest path from T_2 to T_1, and

$$\varepsilon_1=\min\{h(x_{2j+1})\text{: } j=0,1,2,\ldots,l\},$$

$$\varepsilon_2=\min\{b(x_{2j})-h(x_{2j})\text{: } j=1,2,\ldots,l\},$$

$$\varepsilon=\min\{\varepsilon_1, \varepsilon_2\}.$$

Then ε is a positive integer. We modify $h(x)$ as follows:

$$h'(x)=\begin{cases} h(x)-\varepsilon & \text{if } x\in P \text{ and } x=x_j \text{ for odd } j,\\ h(x)+\varepsilon & \text{if } x\in P \text{ and } x=x_j \text{ for even } j,\\ h(x) & \text{otherwise.}\end{cases}$$

Let $w_1'=w_1$ and $w_2'=w_2$. Then a similar argument used in [5] shows that h' is an integer optimum solution to $L_i(h(V')-\varepsilon,\ w_i')$, $i=1,2$. Therefore h' is an integer optimum solution to $L(h(V')-\varepsilon)$.

Case 2: There is no path from T_2 to T_1.

Let B consist of vertices having been reached from T_2 and set

$$\delta_1=\min\{w_1(y)-w_1(x):\ x\in B,\ h(x)>0,\ c_1(D^-_{G_1^*}(S_1(x)))=k,\ y\in S_1(x)\backslash B,$$
$$h(y)<b(y)\},$$
$$\delta_2=\min\{m_1-w_1(x):\ x\in B,\ h(x)>0,\ c_1(D^-_{G_1^*}(S_1(x)))>k\},$$
$$\delta_3=\min\{w_2(y)-w_2(x):\ x\in V'\backslash B,\ \ h(x)>0,\ c_2(D^-_{G_2^*}(S_2(x)))=k,\ y\in S_2(x)\cap B,$$
$$h(y)<b(y)\},$$
$$\delta_4=\min\{m_2\ -\ w_2(x):\ x\in V'\backslash B,\ \ h(x)>0,\ c_2(D^-_{G_2^*}(S_2(x)))>k\},$$
$$\delta=\min\{\delta_1,\delta_2,\delta_3,\delta_4\}.$$

(The minimum is defined to be $+\infty$ when it is taken over the empty set.)

Then $\delta>0$. If δ is finite, we modify w_1 and w_2 as follows:

$$w_1'(x)=\begin{cases} w_1(x)+\delta & \text{if } x\in B,\\ w_1(x) & \text{otherwise,}\end{cases}$$

$$w_2'=w^*-w_1'.$$

A similar argument used in [5] shows that $h'=h$ is an integer optimum solution to $L_i(h(V'),\ w_i')$, $i=1,2$.

Now we apply the procedure again with h', w_1' and w_2'. Then the new $A'\supseteq A$, and, if Case 2 occurs again, $B'\supset B$ as at least one vertex is added to B. Furthermore, $T_i'\supseteq T_i$, $i=1,2$. Consequently, we apply the procedure at most $|V'|$ times, either Case 1 occurs or $\delta=+\infty$. When the latter case occurs, then the current h is an integer optimum solution to L if $h(V')=k-f(V)$; otherwise, G_1 and G_2 have no common root function r with $r(V)=k$, since the algorithm yields two families $\mathscr{F}_1$ and $\mathscr{F}_2$ (possibly empty) of disjoint (nonempty) subsets of V and $I_i\subseteq\bigcup_{S\in\mathscr{F}_i}S$, $i=1,2$, for which (1) does not hold.

Indeed, if $T_1\neq\emptyset$ and $T_2\neq\emptyset$, then, for each $x\in V'$ with $h(x)>0$, either $c_1(D^-_{G_1^*}(S_1(x)))=k$ or $c_2(D^-_{G_2^*}(S_2(x)))=k$. Let $\mathscr{F}_1^*$ consist of all the maximal members in

$$\{S_1(x):\ x\in V',\ h(x)>0,\ c_1(D^-_{G_1^*}(S_1(x)))=k\}$$

and $\mathscr{F}_2$ consist of all the maximal members in

$$\{S_2(x):\ x\in V'\backslash\textstyle\bigcup_{S\in\mathscr{F}_1^*}S,\ h(x)>0,\ c_2(D^-_{G_2^*}(S_2(x)))=k\}$$

(maximal with respect to inclusion). Note that, for each $S\in\mathscr{F}_i^*$, if $v'\in S$, then $v\in S$ as the capacity $c_i(vv')=k$. For $i=1,2$, set

$$\mathscr{F}_i=\{S\cap V\colon S\in\mathscr{F}_i^*\},$$

$$I_i=\{v\in V\colon v'\in \textstyle\bigcup_{S\in\mathscr{F}_i^*}S\}.$$

It is not hard to see that, for each $v\in I_1\cap I_2$, $h(v')=b(v')$ $(=g(v)-f(v))$, yielding $h(v)+h(v')=g(v)$. Simple counting shows that

$$h(V')=$$

$$\sum_{i=1}^{2}\sum_{S\in\mathscr{F}_i}\left\{k-\sum_{v\in S\cap I_i}b_i(D_{\bar{G}_i}^-(v)\cap D_{\bar{G}_i}^-(S))-\sum_{v\in S\setminus I_i}[k-a_i(D_{\bar{G}_i}^-(v)\setminus D_{\bar{G}_i}^-(S))]\right\}$$

$$-g(I_1\cap I_2)-f(I_1\cup I_2),$$

from which it follows that (1) does not hold by using $h(V')>k-f(V)$.

If $T_1=\emptyset$ or $T_2=\emptyset$, say $T_1=\emptyset$, then, for each $x\in V'$ with $h(x)>0$, $c_1(D_{\bar{G}_1}^{-}(S_1(x)))=k$. Let $\mathscr{F}_1^*$ consist of all the maximal members in

$$\{S_1(x)\colon x\in V',\ h(x)>0\}.$$

Set

$$\mathscr{F}_1=\{S\cap V\colon S\in\mathscr{F}_1^*\},$$

$$I_1=\{v\in V\colon v'\in \textstyle\bigcup_{S\in\mathscr{F}_1^*}S\},$$

$$\mathscr{F}_2=\emptyset,$$

$$I_2=\emptyset.$$

Then, similarly, (1) does not hold.

Finally, let us estimate the complexity of the algorithm.

The labelling technique requires at most $|V'|^2$ steps to find a path or the subset B. If Case 2 occurs, the current labels can be used again since $B'\supset B$, $T_1'\supseteq T_1$ and $T_2'\supseteq T_2$. Consequently, if Case 1 has occurred at any time, it will have occurred again after at most $p|V'|^2$ steps, where p denotes the complexity of the max-flow min-cut algorithm. We may assume $g(v)\leqslant k$ for all $v\in V$. Then Case 1 occurs no more than $k|V'|$ times; the algorithm will stop. The output is either a minimum weighted common root function r or two families $\mathscr{F}_1$ and $\mathscr{F}_2$ of disjoints subsets of V and $I_i\subseteq\bigcup_{S\in\mathscr{F}_i}S$, $i=1,2$, for which (1) does not hold. Therefore the complexity of the algorithm is bounded by $O(pk|V|^3)$.

Acknowledgment

The author wishes to thank Prof. Peter L. Hammer for his hospitality and stimulating discussions, and also to thank the referee for his helpful comments.

References

[1] M. Cai, Common root functions of two digraphs, J. Graph Theory 13 (1989) 249–256.
[2] M. Cai, Restricted covering of digraphs with arborescences, Discrete Math. 82 (1990) 143–152.
[3] M. Cai, Covering two digraphs with arborescences, J. Systems Sci. Math. Sci. 3 (1990) 273–277.
[4] J. Edmonds, Edge-disjoint branchings, in: R. Rustin, ed., Combinatorial Algorithms (Academic Press, New York, 1973) 91–96.
[5] A. Frank, A weighted matroid intersection algorithm, J. Algorithms 2 (1981) 328–336.
[6] A. Frank, An algorithm for submodular functions on graphs, Ann. Discrete Math. 16 (1982) 189–212.
[7] A. Frank and E. Tardos, An algorithm for the unbounded matroid intersection polyhedron, Ann. Discrete Math. 19 (1984) 129–134.
[8] L. Lovász, On two minimax theorems in graph theory, J. Combin. Theory Ser. B 2 (1976) 96–103.